ATTENDANCE MANAGEMENT SYSTEM



Suresh Chaudhary

NCC-ID: 00169905

L5DC

Softwarica College of IT & E-Commerce

Dillibazzar, Kathmandu

Contents

[Abstract 6](#_Toc6231202)

[Acknowledgement 6](#_Toc6231203)

[Introduction 6](#_Toc6231204)

[Justification 7](#_Toc6231205)

[Problem Identification 7](#_Toc6231206)

[Aims 8](#_Toc6231207)

[Objectives 8](#_Toc6231208)

[Background to the system 8](#_Toc6231209)

[WBS 9](#_Toc6231210)

[Chapter 2: Analysis 10](#_Toc6231211)

[2.1. Introduction 10](#_Toc6231212)

[2.2. Requirement gathering techniques 11](#_Toc6231213)

[2.4. Analysis methodology 11](#_Toc6231214)

[Hard Approach Methodology 12](#_Toc6231215)

[2.5. SRS (Software Requirement Specification) 14](#_Toc6231216)

[Functional requirement 15](#_Toc6231217)

[Non-functional requirement 19](#_Toc6231218)

[Requirement prioritization 21](#_Toc6231219)

[MoSCoW 21](#_Toc6231220)

[Hardware requirement and Software requirement 23](#_Toc6231221)

[2.7. NLA 24](#_Toc6231222)

[Class Diagram 25](#_Toc6231223)

[Architecture 26](#_Toc6231224)

[Waterfall Model 27](#_Toc6231225)

[Waterfall Model – Design 27](#_Toc6231226)

[Chapter 3 Design 29](#_Toc6231227)

[3.1 Structural Design 29](#_Toc6231228)

[Class diagram 29](#_Toc6231229)

[Flow chart diagram 31](#_Toc6231230)

[3.2 Behavioral Design 34](#_Toc6231231)

[Use Case Diagram 34](#_Toc6231232)

[Activity Diagram 36](#_Toc6231233)

[Sequence diagram 40](#_Toc6231234)

[Data Dictionary 42](#_Toc6231235)

[Entity-Relationship Model 44](#_Toc6231236)

[3.4 User Interface Model 45](#_Toc6231237)

[Prototyping Tool 45](#_Toc6231238)

[Prototype 45](#_Toc6231239)

[Chapter 4: Implementation 48](#_Toc6231240)

[4.1 Programming language 48](#_Toc6231241)

[4.2 Development Environment 48](#_Toc6231242)

[4.3 Development Strategy 48](#_Toc6231243)

[4.4 User Training 49](#_Toc6231244)

[Chapter 5: Testing 50](#_Toc6231245)

[Black Box Testing 50](#_Toc6231246)

[Unit testing 53](#_Toc6231247)

[Chapter 6: Other project issues 55](#_Toc6231248)

[Risk Management 55](#_Toc6231249)

[6.4 Configuration management 56](#_Toc6231250)

[Scheduling 58](#_Toc6231251)

[Future Works 60](#_Toc6231252)

[User Manual 60](#_Toc6231253)

[6.2 Limitations of your project 64](#_Toc6231254)

[Conclusion 64](#_Toc6231255)

[8. References 65](#_Toc6231256)

[Chapter 9: Appendix 66](#_Toc6231257)

[Figure 1:WBS 8](#_Toc6213402)

[Figure 2:DFD level 0 12](#_Toc6213403)

[Figure 3DFD level 1 13](#_Toc6213404)

[Figure 4: Class Diagram 24](#_Toc6213405)

[Figure 5: Architecture 25](#_Toc6213406)

[Figure 6: Waterfall Model 26](#_Toc6213407)

[Figure 7: Final Class Diagram 27](#_Toc6213408)

[Figure 8: Flow Chart Diagram Login 29](#_Toc6213409)

[Figure 9: Flow Chart Diagram 30](#_Toc6213410)

[Figure 10: Use Case Diagram 32](#_Toc6213411)

[Figure 11: Activity Diagram Login 34](#_Toc6213412)

[Figure 12: Activity Diagram 35](#_Toc6213413)

[Figure 13: Sequence Diagram 36](#_Toc6213414)

[Figure 14: ER Diagram 40](#_Toc6213415)

[Figure 15: Unit Test 49](#_Toc6213416)

[*Figure 16 FILE DIRECTORY IN PC* 51](#_Toc6213417)

[*Figure 17 BACKUP FILE DIRECTORY* 51](#_Toc6213418)

[Figure 18: Scheduing 52](#_Toc6213419)

[Figure 19: Scheduling time 53](#_Toc6213420)

# Acknowledgement

I would like to express my greatest thanks of gratitude to my teacher (Mr. Kiran Rana) as well as our Organization who gave me the opportunity to do this project on the topic Attendance Management System. While doing this task I have done a lot of research and I came to know about so many new things I am thankful of them.

Secondly, I would also like to thank my friends who helped me by hanging out with me in finalizing this project within the limited period.

# Abstract

In this project, we present the detailed development and implementation of simple attendance management system. The attendance management system consists of graphical user interface, attendance marker and database: implemented using C++ and Asp.net. The Attendance management is implement using efficient algorithm. The Attendance manager takes attendance of the students entered by the user. This project gives an insight in to the different aspects of Asp.net programing

# Introduction

Attendance management system for managing the attendance of any students or the staffs of any organization. This Management system is mainly for the best and easy way of taking attendance.

## Background to the system

To develop the system for attendance management, the object oriented methodology is useful with their applicability for managing. In Object oriented Software development, the developer identifies and organize the application in terms of needs of object-oriented concepts and by identifying the end users too.

In object oriented software development, there are major phases: they are object-oriented analysis, object oriented design and object oriented implementation and testing.

At the stage of analysis, the developer produces the model of system in the way of their function and how that was to be developed. At the object oriented design system design done in the method of system analysis and architecture, and object design helps in the creation of new class or existing classes for internal details of classes and their associations.

## Justification

The attendance system is developing in the purpose of making the attendance system more efficient, reliable and computerized way of taking attendance at any organizations. This software can also generate the reports about the attendance at the time of needs. Due to the managing retrieval and way of storing data fast and efficient, this software is user friendly. The graphical user interface managed in the purposed way of users to work in the system properly.

This system also can manage the present of any students who may missed their attendance but requests or informed already for the missed classes or attendance can be made as present or update their present condition. In the case of unavailability of the teacher in that lecture, time another staff or lecture can also approve the attendance and get their present condition of their students.

While registering the id of the students is only possible when the id or roll no of their admission fed in the database of the admin. If the teacher registers into the system for id, the admin only can access the permission for their registration and will sent the verification to the admin.

At the time of approving the register requests, then the teacher can login for the taking attendance in their lecture time.

The login section in the system is separate for admin, teachers, the students, and their staffs too. The admin can access the login and registration system of teachers. The teachers can take the attendance, update and register to their students.

Then the students can only view their attendance report and requests for any queries to their lecture teacher.

## Problem Identification

While making attendance system there may be some of the problems listed below:

There may be taking all the presents to the register so that the register may be lost or it may be large to control that type of systematic way.

There may cause some defects such as the un present students may also be taken as present list

In the age of technical system, it’s not the best way to take attendance in the files

## 

## Aims

* To maintain the attendance management system as computerized the traditional way of taking and keeping it.
* As the system of taking this way of attendance, the parents of the students will get SMS automatically about the present of any days
* To develop web application for managing the attendance system of college
* To control the attendance condition for the students and their staff too at the college

## Objectives

* This system is managed to make the attendance management system automated
* This will also help in managing, tracking and updating the data
* To provide the accurate tie periods for the attended staffs
* It will also help in creating the desired reports in short span of time
* To create less amount of proxy attendance

# Chapter 2: Analysis

## 2.1. Introduction

Analysis is the first and the premier advance towards the improvement of a venture. Investigation is the examination and assessment of information or data, by breaking it into its segment parts to reveal their between connections. Examination contemplates constant changes and incorporates joining, measures and points of confinement into a framework. Requirement Analysis is performed to decide client desires for another framework in a nitty gritty and important way. Investigation is an essential part of undertaking the board as it guarantees most extreme convenience of the framework. Diverse examination is performed in the improvement of a framework to stay away from any conceivable issues later on**.**

## 2.2. Requirement gathering techniques

Requirement are the establishment on which the entire task will be based on. The best possible necessity gathering for a product venture relies on the correspondence between the engineer and the partner. The designer never has total data about a business association all alone so legitimate correspondence is the key for the engineer to have the capacity to foresee the necessities by asking the correct inquiries amid the prerequisites gathering period of an undertaking. There are a wide range of necessity assembling methods, for example, pursues:

* Questionnaire
* Interview
* Focus group
* Brainstorming
* Observation
* Prototyping
* JRD (Joint Requirements Development)
* Workshops

Among the techniques mentioned above, I have chosen to use two requirement gathering techniques namely, interview and observation due to the following reasons:

Interview

Interviews are essential methods for social event prerequisites straightforwardly by conversing with the important partner’s vis-à-vis. To have the capacity to impart and tune in to the issue regions and desires for the clients and partners, straightforwardly tending to the prerequisites from the partners makes an incredible programming.

Observation

The analyst watches and finds out about the tasks, process, work process and open doors for enhancements can be resolved. Verifiable prerequisites that are normally disregarded can be revealed. Perception can give a great deal of data from the clients and close partners.

## 2.4. Analysis methodology

Analysis Methodology of components and structure of the framework is analyzed while investigation approach is the example of how we dissect diverse angles for structure a framework and stay away from conceivable issues. Framework examination is the investigation of current business framework (for example PC based) and its issues. Framework examination is useful for critical thinking. There are various methodologies for System investigation, for example,

 Hard approach

 Soft approach

 Combined approach

 Multi-see investigation

 Object-situated investigation

Among the above investigation philosophies, I have utilized hard methodology Analysis procedure for the improvement of Attendance Management System.

## Hard Approach Methodology

Hard System Approach can use to address both subjective and quantitative issues. Hard methodology is task arranged and most appropriate for frameworks that have fixed necessities where the attention is on the improvement of the organization and less on the advancement of the representatives. Representatives ought to be utilized as expense viably and gainfully as could be expected under the circumstances. Motivations to pick hard approach for the examination of Attendance Management System are as per the following:

 follows organized strategies and logical way to deal with critical thinking.

 Suitable for association arranged framework.

 Best appropriate for frameworks that has fixed arrangement of necessities.

 can be utilized to address both subjective and quantitative issues.

## 2.5. SRS (Software Requirement Specification)

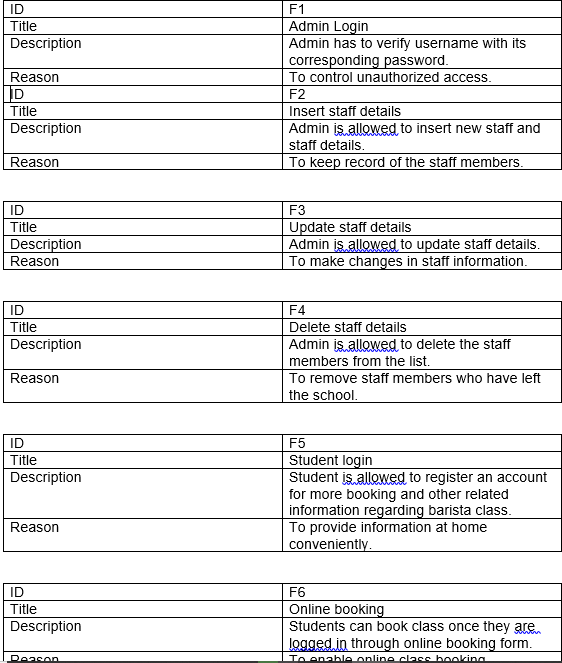
Programming Requirement Specification is an exhaustive portrayal of the planned reason and condition of the framework to create. It incorporates the total depiction about how the framework expected to perform with its utilitarian and non-practical necessities. There is various Software Requirement Specification types, for example, Functional requirement

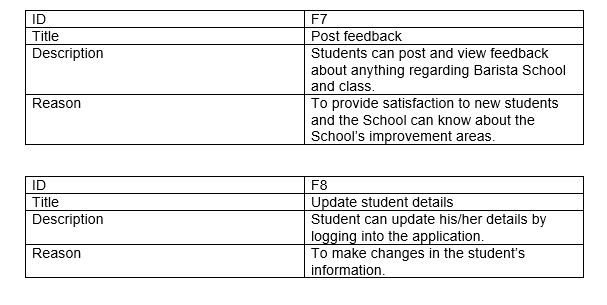
* Non-functional requirement
* Hardware requirement
* Software requirement
* Business requirement
* Market requirement
* User Interface (UI) requirement

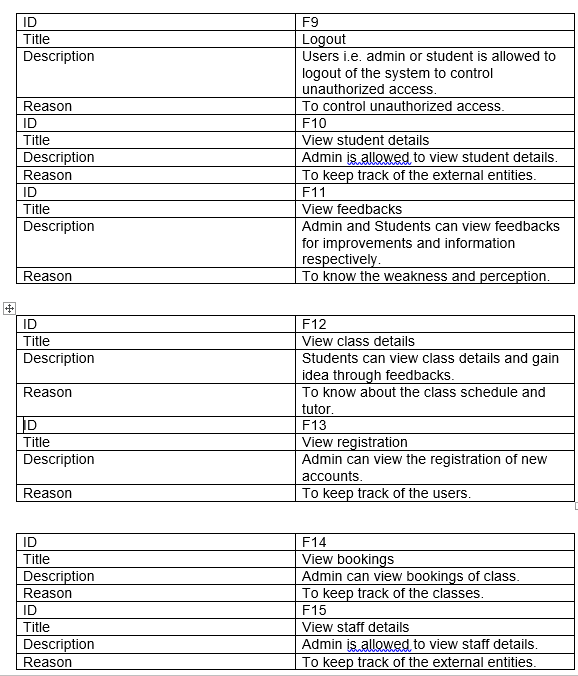
Among the above-mentioned requirements, for the development of Attendance Management System, I have decided to specify the first four requirements i.e.

## Functional requirement

Functional Requirement are those, which identifies with the specialized usefulness of the framework. Practical Requirement manage what the framework ought to do and accommodate the clients. Useful necessity portrays the item highlights and activities with which the client work is associated. Practical necessities bolstered by non-useful requirement. The following are the table that portrays the useful necessities of Attendance Management System:







## Non-functional requirement

Non-functional requirement determines criteria that can be utilized to pass judgment on the activity of a framework specifically conditions, instead of explicit practices. Non-practical necessities characterize how a framework should function and furthermore characterizes framework qualities, for example, execution, security, dependability and the other way around. Non-utilitarian prerequisites are otherwise called quality traits for the framework. The following are ten quality traits for Attendance Management System.

|  |  |
| --- | --- |
| ID | NF1 |
| Title | Security |
| Description | The system should maintain security of the staff and student data. |
| Reason | To prevent loss of data. |

|  |  |
| --- | --- |
| ID | NF2 |
| Title | Performance |
| Description | The system should provide maximum satisfaction for each distinct type of user-computer interaction. |
| Reason | To provide user satisfaction. |

|  |  |
| --- | --- |
| ID | NF3 |
| Title | Scalable |
| Description | The system should be able to accommodate the growing about of work. |
| Reason | To provide flexibility in the future. |

|  |  |
| --- | --- |
| ID | NF4 |
| Title | Recoverable |
| Description | The system should have the ability to prepare and respond to any possible disaster. |
| Reason | To prevent inconvenience. |

|  |  |
| --- | --- |
| ID | NF5 |
| Title | Availability |
| Description | There should not be any unplanned downtimes or outages/ unavailability of the system. |
| Reason | The system should always be available. |

|  |  |
| --- | --- |
| ID | NF6 |
| Title | Maintainability |
| Description | Any faults or problems in the system should be easily fixed and upkeep. |
| Reason | To always upkeep the system effectiveness. |

|  |  |
| --- | --- |
| ID | NF7 |
| Title | Reliability |
| Description | System should assure that there will not be any failure of the system. |
| Reason | To ensure dependability. |

|  |  |
| --- | --- |
| ID | NF8 |
| Title | Usability |
| Description | System should meet the requirements of the user with regards to ease of user. |
| Reason | To allow user-friendliness. |

|  |  |
| --- | --- |
| ID | NF9 |
| Title | Environmental |
| Description | System should be able to operate perfectly in any given environment. |
| Reason | Ensures availability. |

|  |  |
| --- | --- |
| ID | NF10 |
| Title | Data integrity |
| Description | Data integrity ensures that the quality of data has not been exposed or modified. |
| Reason | To prevent data modification. |

## Requirement prioritization

Prioritization can connected to necessities, assignments, items, use cases, test and the other way around. Prerequisite prioritization used to figure out which competitor necessity of a product item ought to be incorporated into a specific discharge. Prioritization causes us distinguish the highlights that is must to need to not important to have. It causes us oversee time and assets for the framework. There are various prioritization methods to be specific:Bubble Sort Technique

## MoSCoW

Ranking

Numerical Assignment (Grouping)

Hundred-dollar method

Analytic hierarchy process

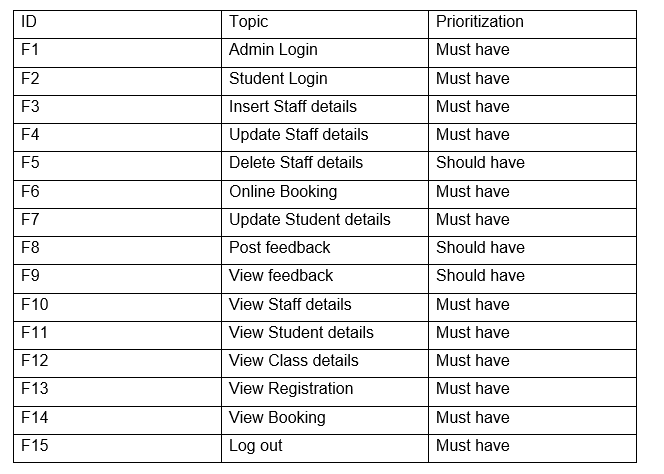
I will be using MoSCoW prioritization technique for Attendance Management System. MoSCoW prioritization originated from the Dynamic Software Development Method (DSDM). MoSCoW prioritization uses four-priority group’s i.e.

Must have (Mandatory)

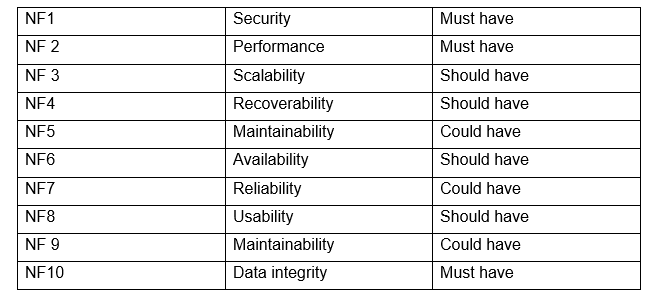
Should have (less vital but important)

Could have (Preferred but not necessary)

Won’t have (Can be postponed and suggested for future execution). MoSCoW prioritization for the **Functional requirements** of Attendance Management System is tabled below:



MoSCoW prioritization of **Non-Functional requirements** tabled below:



## 2.7. NLA

Natural Language investigation (NLA) used to recognize conceivable contender for class, traits and capacities from a given situation. Applicant classes recognized as things, traits as descriptive words and capacities as action words. The following is the situation of Attendance Management System with its NLA separately:

Attendance Management System

The School has reliably been using manual (paper) arrangements until date and the data lost for most by far of the events. The School right now needs to make an online closeness and upgrade their structure from manual to motorized system. The School is hunting down a dealt with structure where the arrangements of classes be done online which will save their time and help them manage the data fittingly. Understudies should have the ability to make class arrangements on the web and incase if they have to get some answers concerning the School, there should similarly be an info fragment where one can end up familiar with about the school in an individual measurement.

The system should have an alternate access to chairperson and client. The understudies should have the ability to book class, see analysis and post input. Clients should in like manner have the ability to invigorate and delete the contribution from one's record. The executive should have the ability to view, update and eradicate the arrangements and staff nuances. Most importantly, the structure should be anything but difficult to utilize and fruitful.

NLA for above situation:

|  |  |  |
| --- | --- | --- |
| Nouns | Adjectives | Verbs |
| Admin, Faculty, student, teacher, feedback, school, class, staff | User-friendly, easy, time-saving, effective | Add, update, delete, view, book, post |

Potential candidate class from the above table are:

|  |
| --- |
| Candidate Class |
| Student  Faculty  Staff  Class |

## 

## Initial Class Diagram

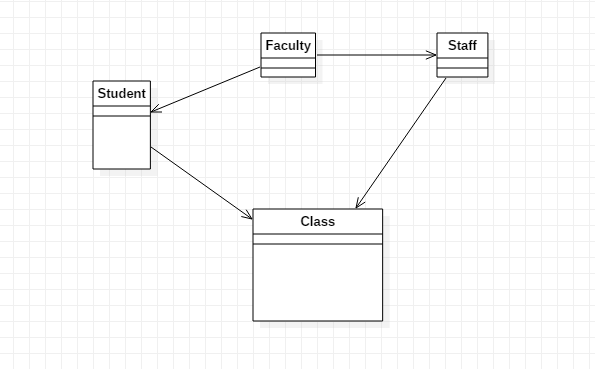


Figure : Initial Class Diagram

### Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system’s users (also known as actors) and their interactions with the system. To build one, you will use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

Scenarios in which your system or application interacts with people, organizations or external systems

Goals that your system or application helps those entities (known as actors) achieve

The scope of your system

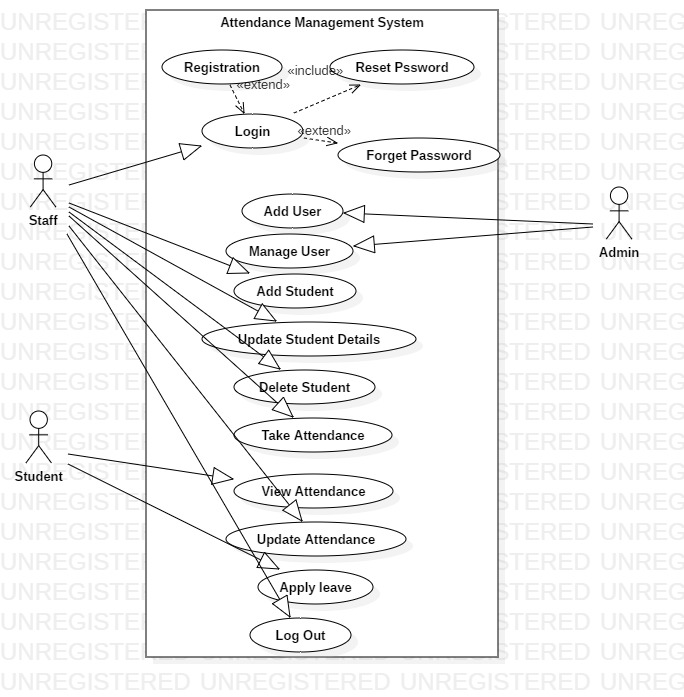


Figure : Use Case Diagram

### Justification

Use case diagram symbols and notation

The notation for a use case diagram is pretty straightforward and doesn’t involve as many types of s symbols as other UML diagrams. Here are all the shapes you will be able to find in Lucid chart:

* Use cases: Horizontally shaped ovals that represent the different uses that a user might have.
* Actors: Stick figures that represent the people actually employing the use cases.
* Associations: A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
* System boundary boxes: A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho killer is outside the scope of occupation in the chainsaw example found below.
* Packages: A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

## Architecture

A 3-level architecture is a sort of programming design, which is made out of three "levels", or "layers" of legitimate registering. They are regularly utilized in applications as a particular kind of customer server framework. 3-level models give numerous advantages to creation and improvement situations by modularizing the UI, business rationale, and information stockpiling layers. Doing as such gives more prominent adaptability to advancement groups by enabling them to refresh a particular piece of an application freely of different parts. This additional adaptability can improve in general time-to-market and diminish advancement process durations by enabling improvement groups to supplant or overhaul free levels without influencing different pieces of the framework.

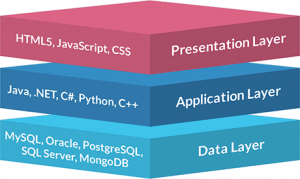


Figure : Architecture

## MVC

## Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

### Waterfall Model – Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

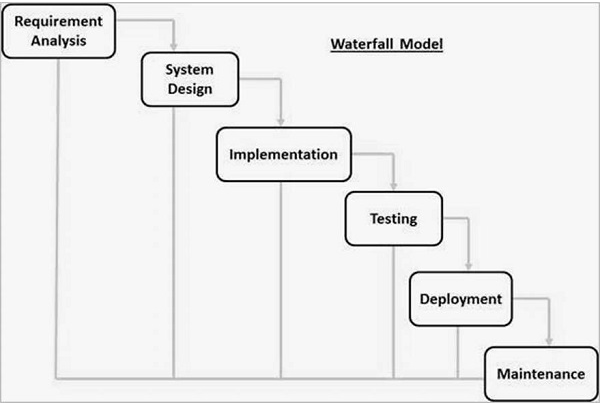


Figure : Waterfall Model

# Chapter 3 Design

**Design** is the creation of a plan or convention for the construction of an object, system or measurable human interaction as in blueprints, drawings, diagrams or patterns.

# Dynamic Modeling

Behavioral diagrams are used to show the dynamic aspects of the system. Behavioral diagrams represent the functionality of a system.

## Sequence diagram

A Sequence diagram portrays association between articles in a successive request for example the request in which these communications happen. We can likewise utilize the terms occasion charts or occasion situations to allude to a sequence diagram. Succession graphs portray how and in what request the items in a framework work.

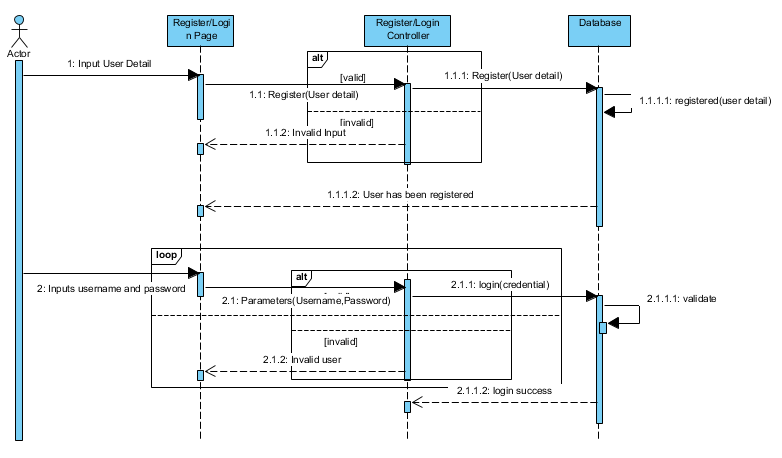


Figure : Sequence Diagram

Basic Sequence Diagram Notations

• Class Roles or Participants

Class jobs depict the manner in which an item will act in setting. UML object image to show class jobs, yet does not list object traits.

• Activation or Execution Occurrence

Actuation boxes speak to the time an article needs to finish an assignment. At the point when an article is occupied with executing a procedure or hanging tight for an answer message, utilize a flimsy dim square shape put vertically on its lifesaver.

• Messages

Messages are permits that speak to correspondence between items. Utilize half-arrowed lines to speak to nonconcurrent message. Nonconcurrent messages sent from an item that will not sit tight for a reaction from the recipient before proceeding with its errands. For message types See underneath

• Lifelines

Life savers are vertical dashed lines that demonstrate the article's quality after some time and

Life expectancy of the article that are taking an interest over a period.

• Destroying Objects

Articles can be ended early utilizing a bolt labeled"<<destroy>>" that focuses to a X. This article expelled from memory. At the point when that object lifesaver closes, you can put a X toward the finish of its help to indicate an annihilation event.

• Loops

A reiteration or circle inside an arrangement outline portrayed as a square shape. Spot the condition for leaving the circle at the base left corner in square sections [ ].

Articles/Participants

- Generally, it is set at the top on the x hub

- Object that start the connection are put at the left

### Activity Diagram

Activity Diagram is another vital chart in UML to depict dynamic parts of the framework. Activity Diagram is a stream graph to speak to the stream starting with one action then onto the next action. The Activity can portray as a task of the framework. The stream can be successive stretched or simultaneous. It manages all kind of stream control by utilizing distinctive components like fork, join, and consolidation choice hub.

Purpose

-Draw the activity flow of a system

-describe the sequence from one activity to another

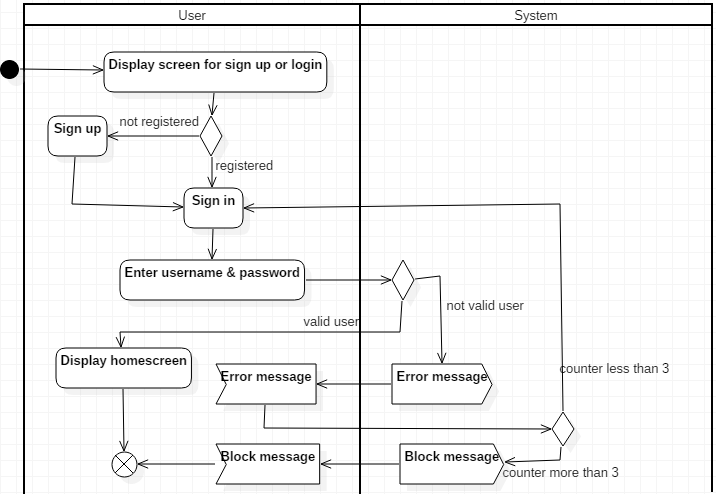
-Describe the parallel, branched and concurrent flow of the system

Figure : Activity Diagram Login

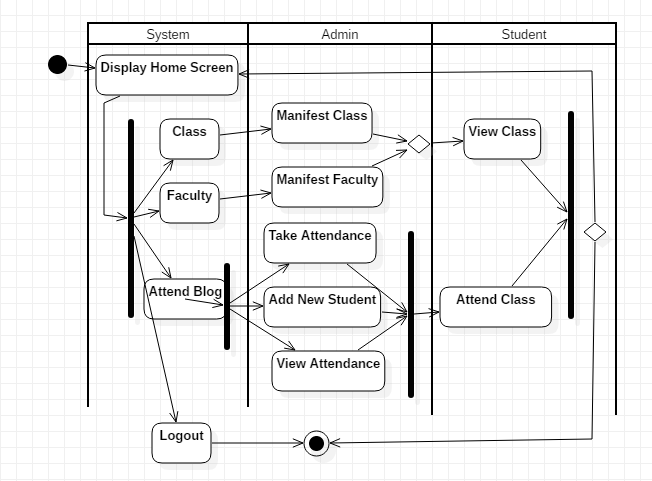


Figure : Activity Diagram

Basic Activity Diagram and Symbols

-Initial state or Start point

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swim lanes, make sure the start point is placed in the top left corner of the first column.

* Activity or Action State
* Action Flow
* Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influence the object. An object flow arrow from an object indicates that the action sate uses the object.

The following diagram depicts a simple Object flow between two actions,

Fill Order and ship order, both accessing order information.

Decision and Branching

A diamond represents a decision with alternate paths. When an activity require a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should labeled with a condition or ground expression. You can also label one of the paths “else”.

# Structural Modeling

Structural Design is a piece of reasonable system that helps constructing and distinguishing essentially productive connection between elements. Auxiliary plan is in charge of the effectiveness of class progressive systems.

### Final Class diagram

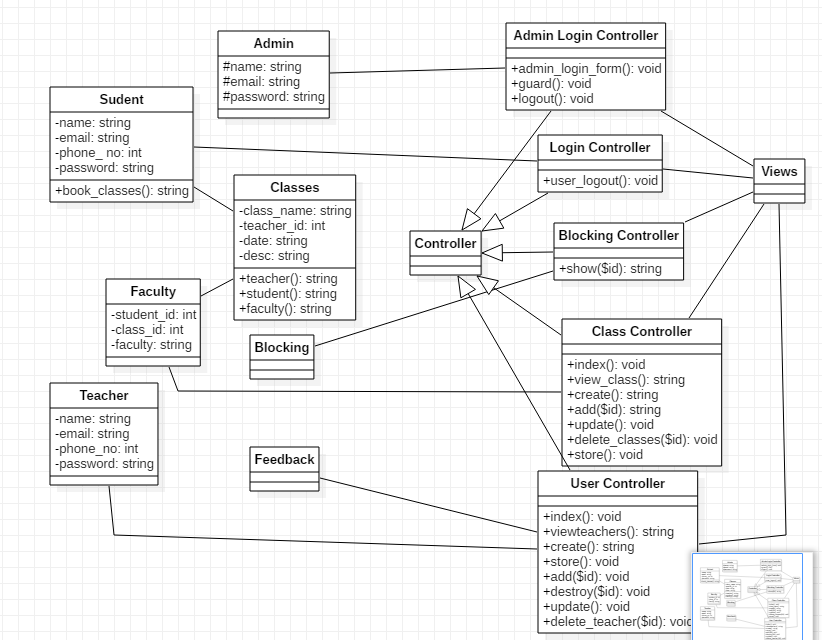
Class diagrams are a standout amongst the most normally utilized basic outlines. Class charts help speak to the static perspective on the framework that is normally object-situated in nature. A class outline made out of classes, affiliations, interfaces and coordinated effort. Class outline helps assemble an executable code for the framework. The following is the class outline for Attendance Management System

Figure : Final Class Diagram

Justification

 Analysis and structure of static perspective on the framework

 Elaborate obligations of framework

 Reverse and forward designing

 It powers the software engineer to thoroughly consider the structure of classes and how they will cooperate with one another before really composing any code. This may prompt an increasingly vigorous application.

 It gives an outline to support to get a review of how the application is organized before analyzing the real code. This may lessen support time.

# DFD

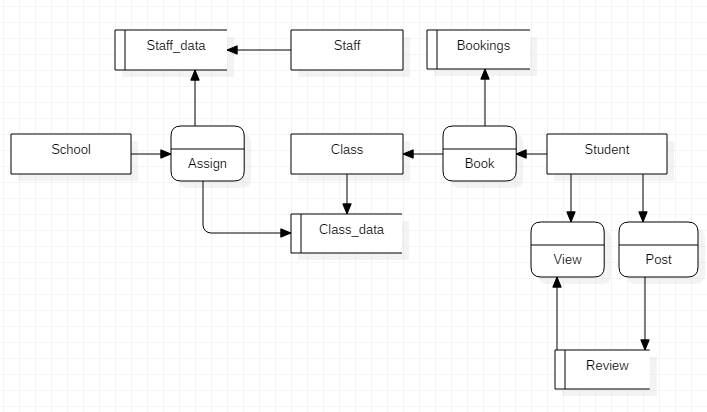
The following are Level 0 and Level 1 Data Flow Diagram (DFD) for Attendance Management System:

Figure :DFD level 0

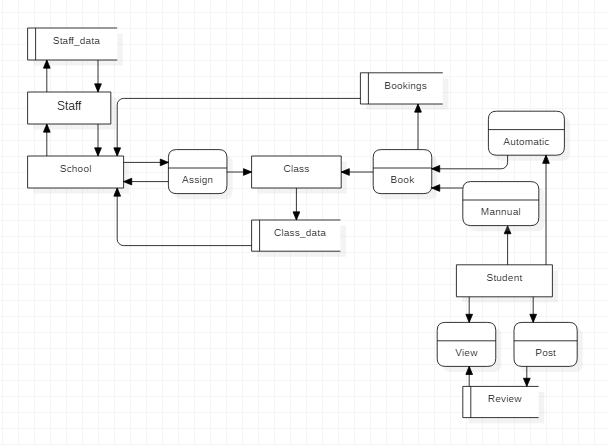


Figure DFD level 1

# Database Modeling

## Entity-Relationship Model

ER (Entity Relationship) diagram is a conceptual design that represents relationships between various entities. Entities are various aspects of the business or project, which we store information about. ER diagram gives the visualization of how the data is stored though out the database system. There is a possibility of having three different relationships in an ER diagram i.e.

* One to one relationship
* Many to many relationship
* Many to one relationship

It provides the blueprint of the actual database. An ER diagram is composed of entities, attributes and connecting lines. Database of the system is designed on the basis of ER diagram. ER diagram for Attendance Management System is given below:

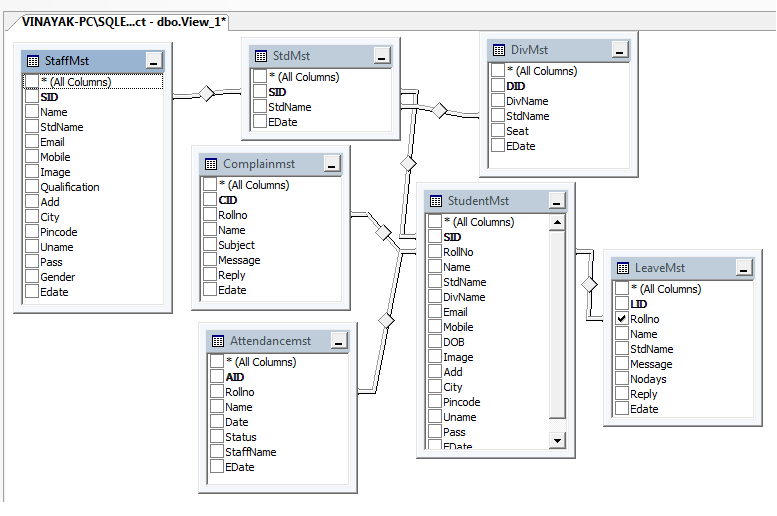


Figure : ER Diagram

## Data Dictionary

Data Dictionary reference is an accumulation of information. Information lexicon comprises of different components, for example, metadata, definition, traits, and information type. Information word reference gives the foundation data of the information. It depicts an information all the more seriously with data about nature of the information. Information lexicon sorts out the dada making it progressively significant. Information word reference is an essential piece of any social database. The following is the information lexicon for Attendance Management System.

Staff Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Column Name | Datatype | PK/FK | Null able |
| 1 | Staff\_ID | Number | PK | Not |
| 2 | Name | Varchar2 | - | Not |
| 3 | DOB | Date | - | Not |
| 4 | Email | Varchar2 | - | Not |
| 5 | Gender | Varchar2 | - | Not |
| 6 | Username | Varchar2 | - | Not |
| 7 | Password | Varchar2 | - | Not |

Client Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Column Name | Datatype | PK/FK | Null able |
| 1 | Student\_ID | Number | PK | Not |
| 2 | Name | Varchar2 | - | Not |
| 3 | DOB | Date | - | Not |
| 4 | Email | Varchar2 | - | Not |
| 5 | Gender | Varchar2 | - | Not |
| 6 | Username | Varchar2 | - | Not |
| 7 | Password | Varchar2 | - | Not |
| 8 | Class\_ID | Number | FK | Not |
| 9 | Usertype\_ID | Number | FK | Not |

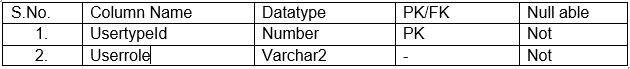
FacultyTable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Column Name | Datatype | PK/FK | Null able |
| 1 | Faculty\_ID | Number | PK | Not |
| 2 | Subject | Varchar2 | - | Not |
| 3 | Class\_ID | Number | FK | Not |

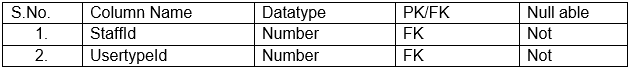
Class Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Column Name | Datatype | PK/FK | Null able |
| 1 | Class\_ID | Number | PK | Not |
| 2 | Time | Varchar2 | - | Not |
| 3 | Fee | Varchar2 | - | Not |
| 4 | Availability | Varchar2 | - | Not |
| 5 | Book | Varchar2 | - | Not |
| 6 | Usertype\_ID | Varchar2 | FK | Not |

Usertype Table



Staff\_Usertype Table



# UI Modeling (Prototype)

The user interface (UI) is the medium through which a user interacts with a system. User interface can include anything from a keyboard to the appearance of the system. User interface allows the input and output of data in the system. It is important for user interface to be attractive and easy to understand in order to make the system user friendly and accessible. Good User interface is an important aspect that defines a good software system.

## Prototype

Model is premise of things to come models. Model helps fill in as reference while planning the real programming framework. Customer prerequisites drawn into a model and the genuine framework grew just when the customer is happy with the model. Model helps manufacture the framework quicker and productively. It makes the improvement of the framework quicker and adaptable. Visualizing the structure of a framework is conceivable with the assistance of model. The following is the model for the proposed administration framework.

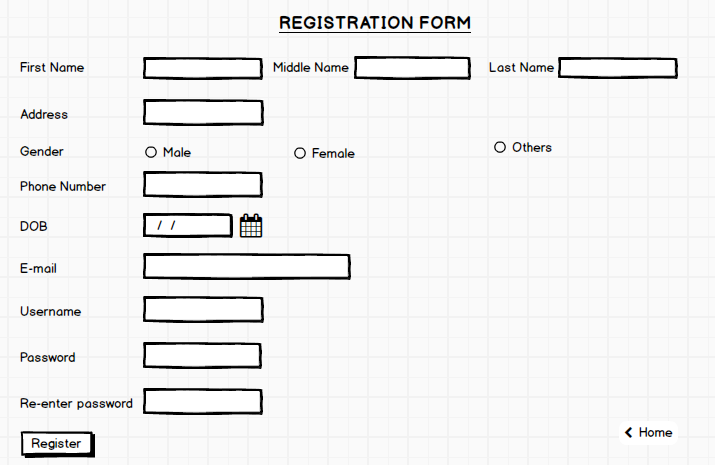


Figure : Prototype register

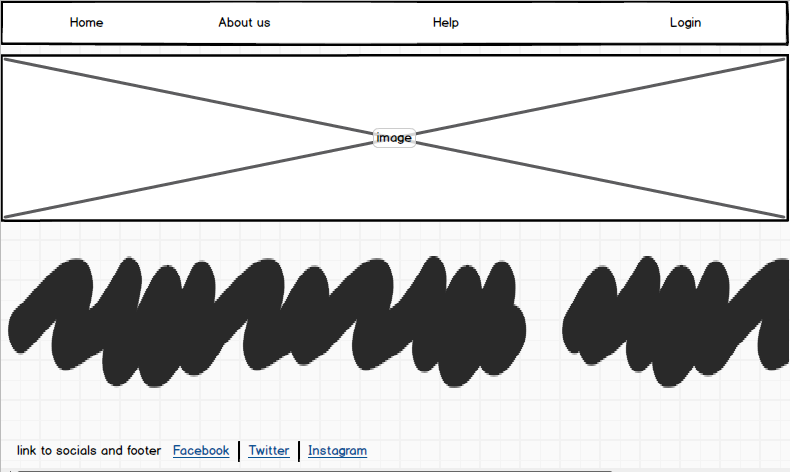


Figure : Prototype Home

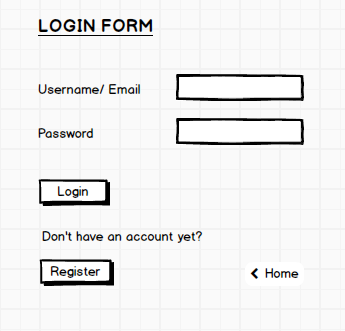


Figure : Prototype Login

# Chapter 4: Implementation

## 4.1 Programming language

Programming language is medium that permits the correspondence between PC and human. Programming dialects enable us to offer directions to a PC in a language that is comprehended by the PC. Like how human dialects are of numerous kinds, there are likewise various programming dialects. Java, C, C++, Php, C#, Ruby are a portion of the basic programming dialects.

For the advancement of Attendance Management System, I have Used item arrange Asp.net for the backend improvement joined with Bootstrap and C++. With respect to the system, I have utilized Asp.net. C++ is a server scripting language that is utilized to create dynamic sites. Php is valuable in making sites progressively intuitive. Php contents are executed on the server.

## 4.2 Development Environment

Real advancement condition for this framework depends on programming language Asp.net. This venture created utilizing numerous diverse devices to finish the task. The devices utilized in the advancement of Attendance Management System are recorded beneath:

Asp.net is the major scripting language for this project.

Visual paradigm is used for creating Entity Relationship diagram and sequence.

Star UML is used for creating activity diagram, use case diagram and data flow diagram.

## 4.3 Development Strategy

Attendance Management System can be set up on a local computer with the visual studio application. For that database, Sql Server Management studio is used for the application. Input and output request is send to the local server.

## 4.4 User Training

The front-end part of this proposed system is basic and easy to understand. User manual is also provided so one does not need to take any extra training to access this website. The system is user friendly and accessible to the users. Any occurrence of confusions can be solved though the user manual or one can also refer to FAQs.

## Hardware requirement and Software requirement

The most common set of requirements defined by a software application is a physical computer. Hardware a software may vary depending on the physical computer and operating system.

* Hardware

The minimum hardware requirement to install and effectively operate Attendance Management System are:

* Laptop/PC
* Processor Intel Core I5
* 4 GB RAM (physical)
* Memory 1TB
* Software

The minimum software requirement to install and effectively operate Attendance Management System are:

* Operating system Windows 10
* SQL Server Management Studio
* Visual Studio 2017
* Web browser (Google chrome preferred)

**UI Screenshots are provided in Appendix**

# Chapter 5: Testing

Software testing is an activity that helps check if the system works correctly and provides expected results. Software testing ensure that the system is free from errors. Testing also, helps detect errors so the bugs or defects can be fixed before it is actually released. It can also define as the process of evaluating a system whether it fulfills the user requirement or not.

There are many different types of software testing methods such as

* Black box testing
* White box testing
* Integration testing
* Regression testing
* Interface testing
* User acceptance testing. To test Attendance Management System, I have chosen the following types of testing:

## Black Box Testing

Black box testing is a technique for programming testing that analyzes the usefulness of an application without peering into its inner structures or operations. This strategy for test can connected to essentially each dimension of programming testing: unit, coordination, framework and acknowledgment. It commonly contains most if not all more elevated amount testing, yet can rule unit testing also.

Testing for Attendance Management System

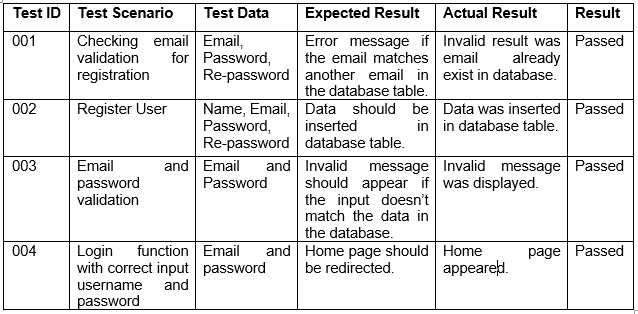
Test date: 18:10:2018

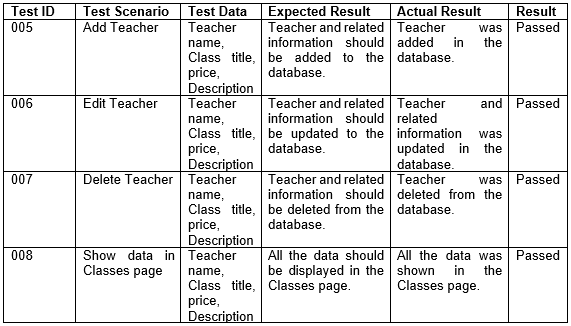
Tester: Suresh Chaudhary

Address: Tulsipur, Dang

Test name: Login and Registration

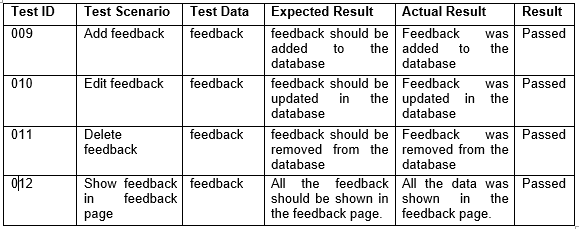
Test ID: 001-004

Test name: Add/Update/Delete Teacher’s data

Test ID: 005-008

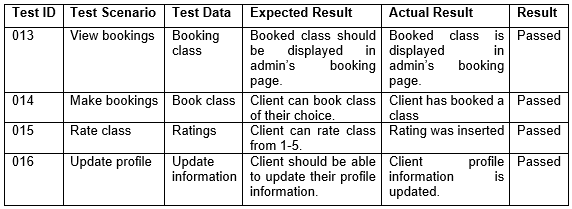
Test name: Add/Update/Delete Feedback

Test ID: 009-012



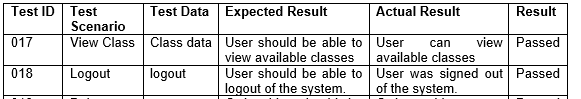
Test name: Add/ View bookings, update profile, rate class

Test ID: 013-016



Test name: View class, Logout

Test ID: 017-018



## Unit testing

Unit testing test a software at unit level. Any part of the software with a smallest testable is a unit. Unit testing has performed using White Box Testing. Unit testing can be done either manually or automatically. In unit testing, units of source code, procedures and data are tested to determine if the system is usable or not. This test focus on the technical details of a system allowing the bugs and errors to be identified early. Unit testing allows changes to be adapted early and easily. Once a system passes unit test, the system considered to complete. Below are the unit tests for Barista School Management System (Reese, Unit Testing 2019).

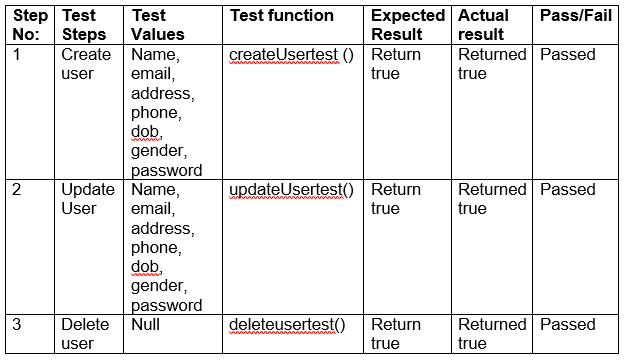
**Unit Testing for Attendance Management System**

Test date: 18:10:2018

Tester: Suresh Chaudhary

Address: Tulsipur

Test log



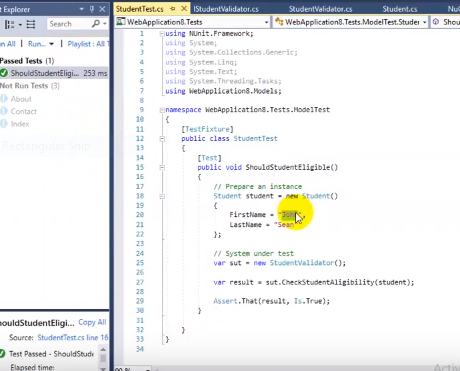


Figure : Unit Test

# Chapter 6: Other project issues

## Risk Management

There may be any types of risks in any management system. To find out the risks we have to analyze all types of works. To do some solutions for the risks we have to keep actions for the risks. Here we have analyzed the risks.

To estimate the impact of each identified risks to the organization we use

Below is the table that shows the likelihood with respect to its value:

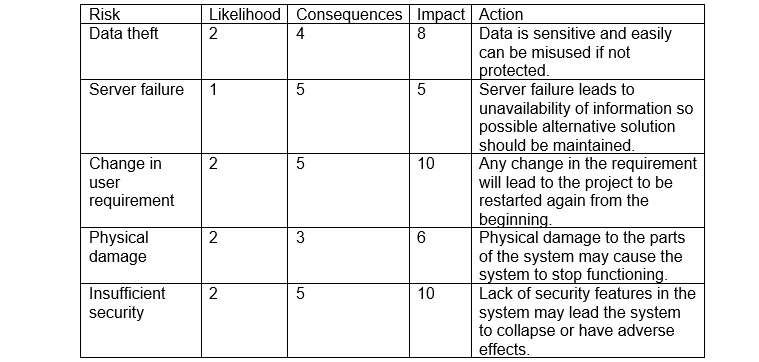
|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Below is the table that shows the consequences with respect to its value:

|  |  |
| --- | --- |
| Consequences | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

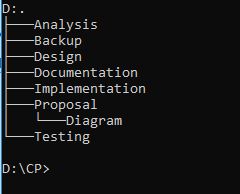
**Impact = Likelihood \* Consequence**

Relation, in this relation the values for Likelihood and Consequence we have find out through the given tables respectively:



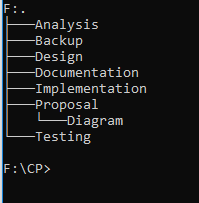
## 6.4 Configuration management

Arrangement the board alludes to efficiently overseeing changes occurring in a framework after some time in way that the uprightness kept up. Arrangement the executives stores every one of the information, documents and data assets in a solitary organizer over an index. Setup the board tends to the organization of a venture, the documentation characterizing it, and other information supporting it. Arrangement the executives portrays the things that make up a framework or programming. Design the board of Attendance executive’s framework:



*Figure 19 FILE DIRECTORY IN PC*

File backup maintained in another drive as shown in the picture below:



*Figure 20 BACKUP FILE DIRECTORY*

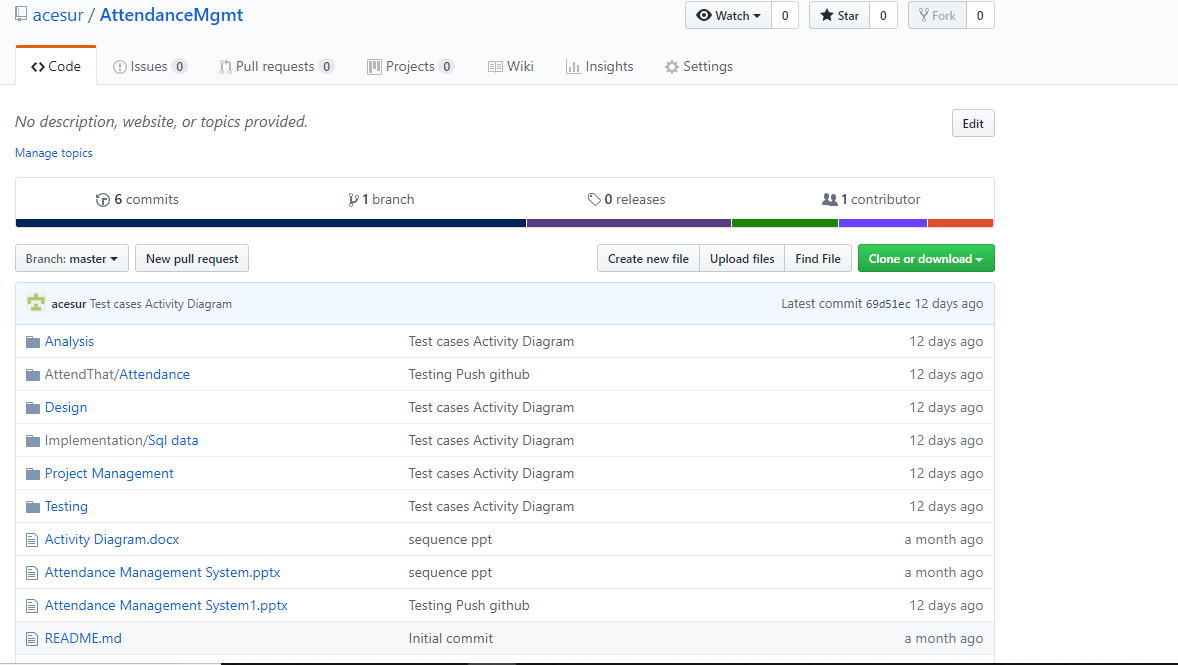


Figure : Git hub

Implementation SS

# Scheduling

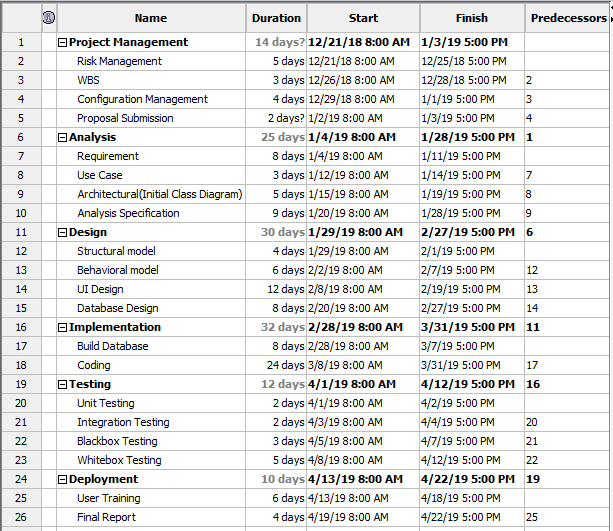


Figure : Scheduling

# Gantt Chart

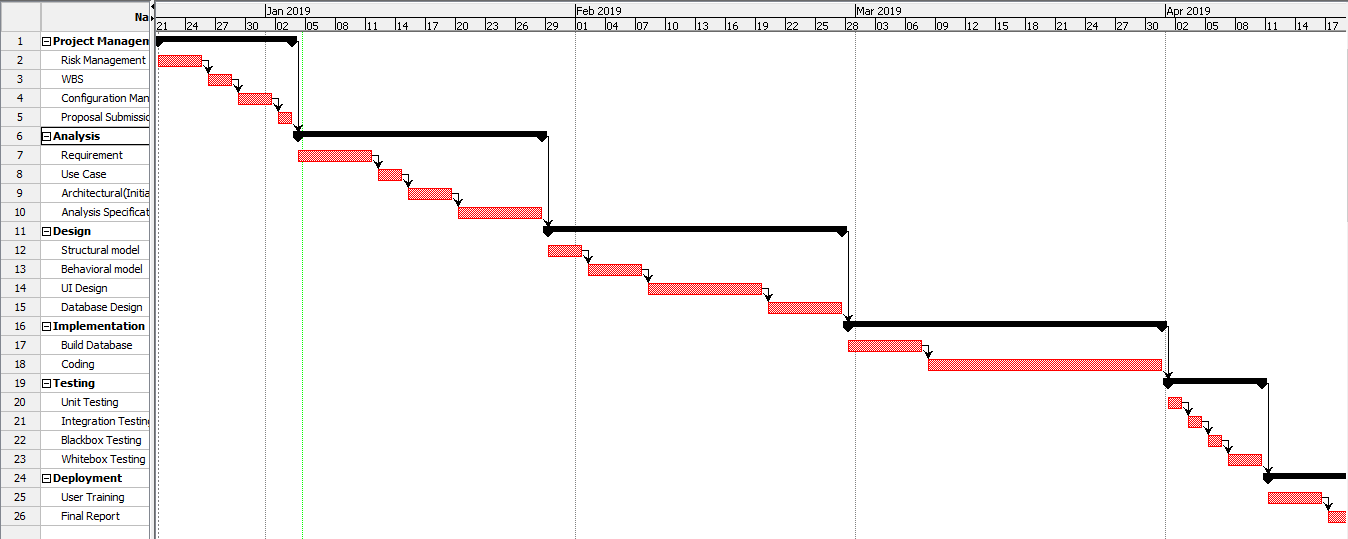


Figure : Scheduling time

## Future Works

In the future, the following features planned to integrate in the system:

* Payment gateway

Payment gateway will provide more convenience to the users. Currently it was not possible to integrate payment gateway in the current project due to limited budget but hopefully it will be integrated in the future.

* Social Login

Social login will integrate in the system in the near future. Social login will allow the users to login through the common social accounts.

* Email notification

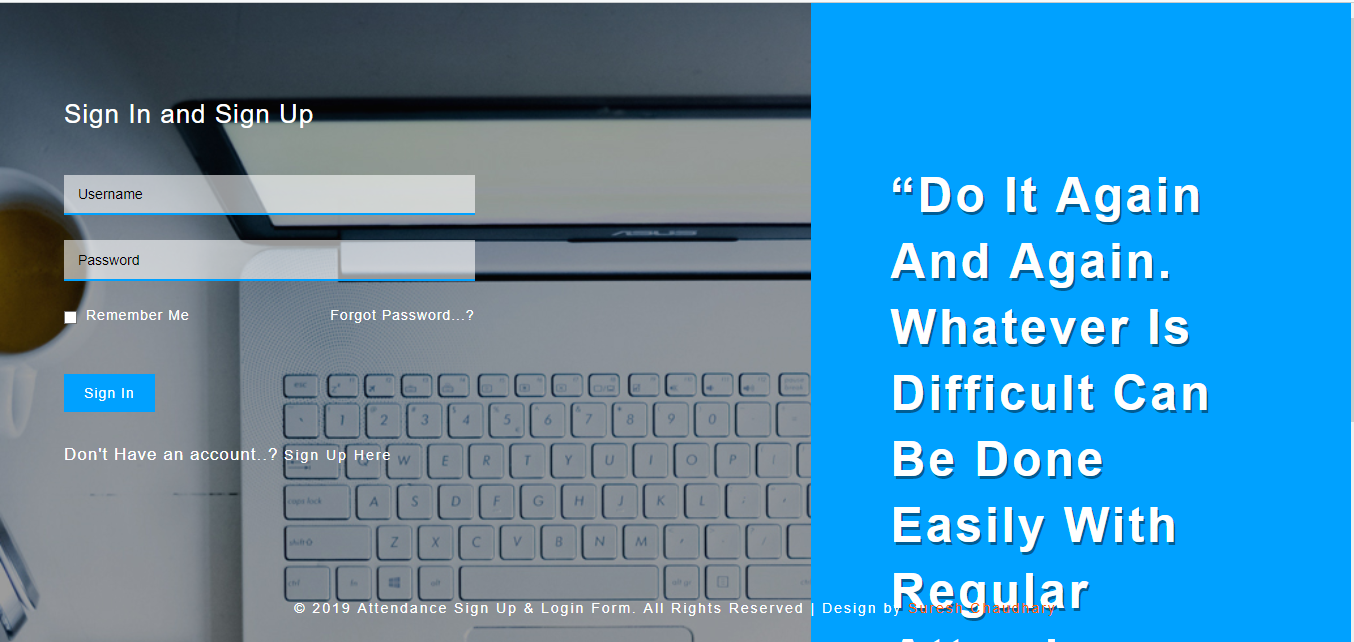
Email notification will integrate in the future when we will be financially stable to buy the mail server.

* Attendance system

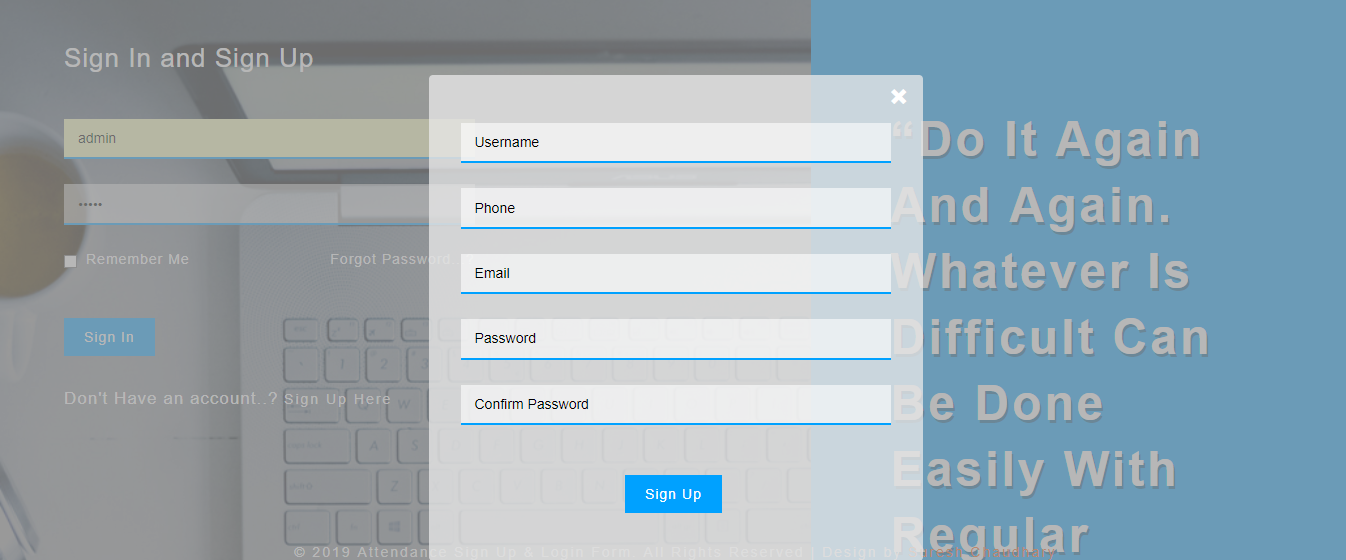
Attendance system will integrate in the system in the future.

# User Manual

For the user to make the system easy to run there are some ways to described how it works. There are some user manual description.

* For login
* Click on login
* First input username and password
* 
* For Register
* click on register

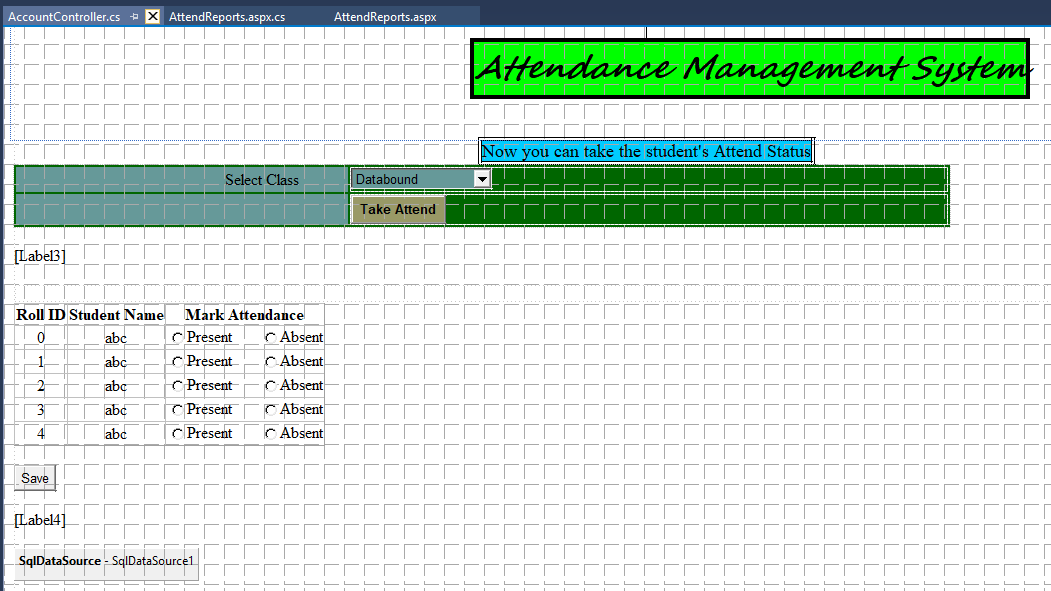
First input all boxes for the description



For taking Attendance

* Click on save

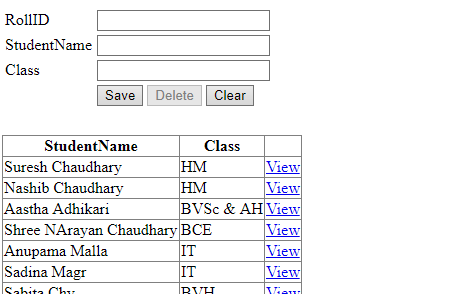
First, choose the subject



Then Choose present or absent and mark

* For adding Student

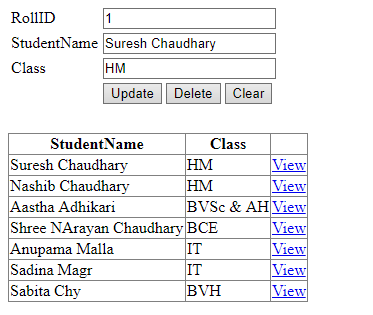
- Click on add



Fill of the student details and Subject

* For updating click on view of student details

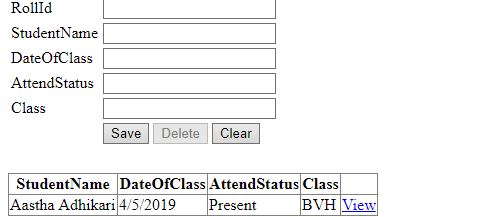
Then Click on update by updating the student



* For Student attendance report update

First click on view

Then Update



## 

## 6.2 Limitations of your project

• Email:

Email warning not incorporated in the framework because of inaccessibility of mail server.

• Payment door:

Installment door not incorporated in the framework because of constrained spending plan.

• Social login:

Login from social locales not incorporated in the framework because of time limits.

# Conclusion

Attendance management system provides to take attendance in the best way. To develop the system object oriented Methodology has chosen as the best methodology for this project. For this project work break down System have also created and submission for task milestones been created.

All the tasks scheduled for the best result.

In addition, to manage the risks, all the processes analyzed and found the impacts of risks by creating the values of likelihood and consequence relation.

# 8. References

Bell, Donald. 2016. *UML sequence diagram.* Accessed 9 19, 2018. https://www.ibm.com/developerworks/rational/library/3101.html.

billows, r. 2017. *Work break down structure.* Accessed 07 10, 2018. https://4pm.com/2017/03/01/work-breakdown-structure-wbs-2/.

Mukund. 2018. *Feasibility study in project management.* Accessed 10 09, 2018. https://www.simplilearn.com/feasibility-study-article.

Rajkurmar. 2018. *Black Box Testing.* Accessed 10 02, 2018. https://www.softwaretestingmaterial.com/black-box-and-white-box-testing/.

Reese, John. 2018. *Unit Testing.* Accessed 10 05, 2018. https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices.

—. 2019. *Unit Testing.* Accessed 09 05, 2019. https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices.

Stafford, Will. 2015. *Importance of requirement gathering.* Accessed 07 27, 2018. https://www.ktlsolutions.com/project-management/the-importance-of-requirements-gathering-for-software-projects/.

# Chapter 9: Appendix

